P⊛RTAL
USPTO

safety abort bytecode emulation

SEARCH

THE ACM DIGITAL LIBRARY

📣 Feedback Report a problem Satisfaction survey

Terms used **safety** **abort** **bytecode**

Found **1,413** of **199,787**

Sort results by	relevance ⬛
Display results	expanded form ⬛

📗 Save results to a Binder
🔖 Search Tips
☐ Open results in a new window

Try an Advanced Search
Try this search in The ACM Guide

Results 1 - 20 of 200
Best 200 shown

Result page: **1** 2 3 4 5 6 7 8 9 10   next

Relevance scale ☐ ⬜ ◰ ◼ ⬛

1 Dynamic languages symposium chair's welcome: Hardware tansactional memory support for lightweight dynamic language evolution ⬛

Nicholas Riley, Craig Zilles

October 2006 **Companion to the 21st ACM SIGPLAN conference on Object-oriented programming systems, languages, and applications OOPSLA '06**

Publisher: ACM Press

Full text available: 📄 pdf(325.22 KB)   Additional Information: full citation, abstract, references, index terms

> Lightweight dynamic language runtimes have become popular in part because they simply integrate with a wide range of native code libraries and embedding applications. However, further development of these runtimes in the areas of concurrency, efficiency and safety is impeded by the desire to maintain their native code interfaces, even at a source level. Native extension modules' lack of thread safety is a significant barrier to dynamic languages' effective deployment on current and future multic ...

> **Keywords**: Python, concurrency, dynamic languages, locking, safety, transactional memory

2 Model-carrying code: a practical approach for safe execution of untrusted applications ◼

R. Sekar, V.N. Venkatakrishnan, Samik Basu, Sandeep Bhatkar, Daniel C. DuVarney

October 2003 **ACM SIGOPS Operating Systems Review , Proceedings of the nineteenth ACM symposium on Operating systems principles SOSP '03**, Volume 37 Issue 5

Publisher: ACM Press

Full text available: 📄 pdf(301.30 KB)   Additional Information: full citation, abstract, references, citings, index terms

> This paper presents a new approach called *model-carrying code* (MCC) for safe execution of untrusted code. At the heart of MCC is the idea that untrusted code comes equipped with a concise high-level model of its security-relevant behavior. This model helps bridge the gap between high-level security policies and low-level binary code, thereby enabling analyses which would otherwise be impractical. For instance, users can use a fully automated verification procedure to determine if the code ...

> **Keywords**: mobile code security, policy enforcement, sand-boxing, security policies

P⊛RTAL
USPTO

THE ACM DIGITAL LIBRARY

▐ Feedback Report a problem Satisfaction survey

Terms used **safety abort bytecode emulation**   Found **2,266** of **199,787**

Sort results by | relevance ▮ | ◆ Save results to a Binder | Try an Advanced Search
Display results | expanded form ▮ | ⓘ Search Tips | Try this search in The ACM Guide
 | | ☐ Open results in a new window |

Results 1 - 20 of 200   Result page: **1** 2 3 4 5 6 7 8 9 10   next
Best 200 shown   Relevance scale ☐▱▰▰▰

1   Dynamic languages symposium chair's welcome: Hardware tansactional memory support for lightweight dynamic language evolution
Nicholas Riley, Craig Zilles
October 2006 **Companion to the 21st ACM SIGPLAN conference on Object-oriented programming systems, languages, and applications OOPSLA '06**
Publisher: ACM Press
Full text available: 🗎 pdf(325.22 KB)   Additional Information: full citation, abstract, references, index terms

> Lightweight dynamic language runtimes have become popular in part because they simply integrate with a wide range of native code libraries and embedding applications. However, further development of these runtimes in the areas of concurrency, efficiency and safety is impeded by the desire to maintain their native code interfaces, even at a source level. Native extension modules' lack of thread safety is a significant barrier to dynamic languages' effective deployment on current and future multic ...
>
> **Keywords:** Python, concurrency, dynamic languages, locking, safety, transactional memory

2   Real-time convergence of Ada and Java™
Ben Brosgol, Brian Dobbing
September 2001 **ACM SIGAda Ada Letters , Proceedings of the 2001 annual ACM SIGAda international conference on Ada SIGAda '01**, Volume XXI Issue 4
Publisher: ACM Press
Full text available: 🗎 pdf(191.98 KB)   Additional Information: full citation, abstract, references, citings, index terms

> Two independent recent efforts have defined extensions to the Java platform that intend to satisfy real-time requirements. This paper summarizes the major features of these efforts, compares them to each other and to Ada 95's Real-Time Annex, and argues that their convergence with Ada95 may serve to complement rather than compete with Ada in the real-time domain.
>
> **Keywords:** Ada, Java, Real-Time, asynchrony, garbage collection, scheduling, threads

3   JRes: a resource accounting interface for Java
Grzegorz Czajkowski, Thorsten von Eicken